liuyc-notify使用文档

# 1、简介

liuyc-notify是一个基于nw-notify实现的可设置通知显示模式的多模板通知模块。我们可以自定义通知的内容和样式，但建议保持模板的宽度相同，如果是有不同宽度的需求，可以通过修改代码中的nextInsertPos.x来设置。还可以设置通知显示的最大数量，且保证通知不会超出屏幕。通知模式目前是有4种：

1、有队列的堆叠：

可以通过setConfig来设置maxVisibleNotifications显示通知最大数量。当通知到达最大数量时或者超出屏幕时，即使有新通知进来，也要等旧通知关闭才有位置显示出来，而不是强制使旧通知结束，这就是有队列的意思。

2、有队列的相同模板通知不替换：

假如a是模板，当a1显示时，又发起a2通知，那么a2要等a1结束才会显示而不会强制a1结束，就会产生队列。

3、相同模板通知替换：

有两种替换方式：分别是在原位置替换通知的内容和先关闭旧通知再在最顶显示新通知。

在原位置：假如a是模板，当a1显示时，又发起a2通知，会把a1通知的内容替换为a2通知的内容，并重新设置持续时间。

在最顶位置：当a1显示时，又发起a2通知，那么a2会强制a1结束，然后a2插队显示出来，就是替换的意思，只会替换最早的一条同模板通知。插队是指就算队列有其他通知，也让这替换通知先显示出来。

但由于增加鼠标悬停不关闭通知的功能，所以当鼠标悬停在a1上时，a2也不能强制a1结束或替换a1的内容，a2进入队列。还有一种情况，当没有可替换且达到最大显示数量的时候，还是会进入队列。

4、相同模板通知舍弃：

同理，假如a是模板，当a1显示时，又发起a2通知，那么a2会舍弃；直到没有a类通知的时候，才能接收a类通知。

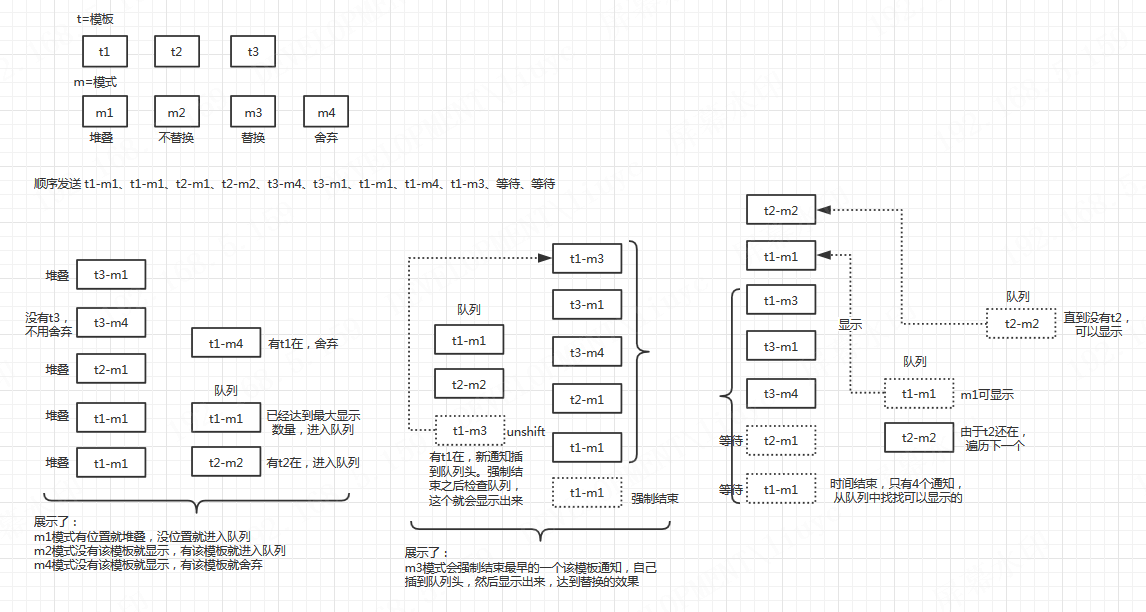


图 1 示例图

# 2、使用

## 2.1、目前模块暴露了5个函数供使用：

1、init：接收一个gui参数，主要是用来初始化gui变量，然后用gui变量处理一些初始化的事情。

**var *gui*** = require(**'nw.gui'**);  
 **var *liuyc*** = require(**'liuyc-notify'**);  
 ***liuyc***.init(***gui***); //必须在一切开始前执行一次

2、setTmplInfo：接受两个数组参数，分别是模板路径数组和窗口高度数组。如果模板是不同宽度的还得加上宽度处理。

//要用到的模板的路径数组 var *tmplArr* = ['html/html/notification.html', 'html/html/notification2.html', 'html/html/notification3.html'];  
//（窗口）模板的高度，按顺序对应，这里我是采用html的height加上padding\*2，padding用于处理mouseleave var *tmplHeightArr* = [60, 140, 90];  
 *liuyc*.setTmplInfo(*tmplArr*, *tmplHeightArr*); //（必需），把函数暴露出来，改变模板时不应对代码进行修改，而是通过设置修改

3、setConfig：用来配置一些通知信息，可查看模块中notify.js文件中的config变量得到可以设置的信息。一般是设置displayTime、appIcon、maxVisibleNotifications为主。

liuyc.setConfig({  
 **displayTime**: 6000, //持续时间，所有通知公用，也可以为通知单独设置  
 **appIcon**: **'../img/icon48.png'** });

4、notify：发起通知，参数tmpl（模板路径，如html/html/notification3.html）是必要的，值必须在tmplArr中包含有的。目前有三个事件，其余的参数是根据模块中notify.js文件中的setNotficationContents函数相互设置，还有displayTime为该通知设置持续时间。以下是示例使用到的属性，也可以在setNotficationContents中设置更多的内容。

*liuyc*.notify({  
 title: 'title',  
 text: '文本',  
 image: '../img/icon48.png', //通知右边的图片  
 url: 'http://www.baidu.com', //使用默认浏览器打开网站，而不是在nw中打开bottomText: '文本',  
 **mode**: 3, //目前可设置1、2、3、4对应简介的4种模式。3是替换，多加一个参数决定是在原位置替换内容，还是关闭通知在新位置显示新通知  
 **showInOld**: **true**, //替换旧通知的内容（false=最顶显示），其他模式不需要这个参数  
 tmpl: 'html/html/notification3.html', //模板的路径，用于打开窗口和比较onClickFunc: function(event){  
 console.log('User clicked notification ' + event.id + '. Closing it immediately.');  
 event.closeNotification();  
 },  
 onShowFunc: function(event) {  
 console.log('onShow ' + numPrivate, event)  
 },  
 onCloseFunc: function(event) {  
 console.log('onClose ' + numPrivate, event)  
 }  
 });

源码中还具有sound声音设置，没有测试过是否可行。

5、closeAll：关闭所有的通知。

liuyc.closeAll();

可以把setNotficationContents暴露出来由用户设置通知内容。也可以按需暴露更多的自定义函数，如返回当前通知数量…

## 2.2、新增模板

1、新建一个html文件，拷贝以下内容到文件中，该文档中不提供css代码，可以参考项目样式文件。

<**html** id=**"html"**>  
 <**head**>  
 <**meta** charset=**"UTF-8"**>  
 <**link** rel=**"stylesheet"** href=**"../css/notification.css"**>  
 </**head**>  
 <**body** style=**'**overflow: **hidden**; -webkit-user-select: **none**; margin: 0;**'**>  
 <**div** id=**"container"**>  
 <**div** style=**"**display: **flex**;**"**>  
 <**img** src=**""** id=**"appIcon"** />  
 <**div** id=**"text"**>  
 <**b** id=**"title"**></**b**>  
 <**p** id=**"message"**></**p**>  
 </**div**>  
 <**img** src=**""** id=**"image"** />  
 </**div**>  
 <**div** id=**"close"**>X</**div**>  
 </**div**>  
 </**body**>  
 </**html**>

模板只要具有container、appIcon、close的id就可以，布局自己设定，如果不是这些id，需要到模块中notify.js文件中搜索getElementById修改对应的id。

2、填充内容，在notify.js文件中的setNotficationContents函数中模仿已有代码设置模板内容。

3、发送通知前调用一次setTmplInfo方法，设置好模板的路径和显示该模板的窗口高度。

# 3、缺陷

目前liuyc-notify模块已发现存在的问题包括：

1、无法设置通知窗口的focus，就是当显示一个通知的时候，通知会自动focus，导致主窗口失焦，这时如果主窗口在进行打字，会受到很大影响。解决的方法是当主程序最小化或托盘时才弹出通知，其他情况由主程序接收通知并进行处理。

2、可以修改源码中的点击事件，应该点击通知回到主程序并进行某些处理才合适，而不是用默认浏览器来处理url。

3、多次测试在Windows 7 Basic主题下，通知关闭后仍会留下当前画面在桌面上，给人一种还没关闭的感觉。正是由于这种画面残留的现象，所以有时通知在下降的时候，会有一种通知位置不对的感觉，可以刷新桌面清除残留的画面。